UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No 0704-0188

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b. RESTRICTIVE MARKINGS: None
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT: Report is approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S): CAA-RP-87-3	5. MONITORING ORGANIZATION REPORT NUMBER (S)
6a. NAME OF PERFORMING ORGANIZATION: US Army Concepts Analysis Agency — 6b. OFFICE SYMBOL (if applicable)	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code): 8120 Woodmont Avenue, Bethesda, MD 20814-2797	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION: US Army Concepts Analysis Agency — 8b. OFFICE SYMBOL (if applicable): CSCA-MVM	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code): 8120 Woodmont Avenue, Bethesda, MD 20814-2797	10. SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO

11. TITLE (Include Security Classification)
An Algorithm for Determining the Location of Points in Computer Simulations

12. PERSONAL AUTHOR(S)
Dr. Robert L. Helmbold

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Research Paper	FROM 87/09 TO 87/10	87/10	46

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Wargaming, simulating, computer programs

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This paper describes a new and effective algorithm for determining in computer simulations the locations of points relative to prespecified boundary or phase lines. The algorithm is widely applicable, explicitly based on general principles, and user friendly.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT: ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL: ROBERT L. HELMBOLD	22b. TELEPHONE (Include Area Code): (301) 295-1669 — 22c. OFFICE SYMBOL: CSCA-MVM

RESEARCH PAPER
CAA-RP-87-3

AN ALGORITHM FOR DETERMINING THE LOCATION OF POINTS IN COMPUTER SIMULATIONS

OCTOBER 1987

PREPARED BY
DR. ROBERT L. HELMBOLD
OFFICE OF SPECIAL ASSISTANT
FOR MODEL VALIDATION

US ARMY CONCEPTS ANALYSIS AGENCY
8120 WOODMONT AVENUE
BETHESDA, MARYLAND 20814-2797

DISCLAIMER

The findings of this report are not to be construed as an official Department of the Army position, policy, or decision unless so designated by other official documentation. Comments or suggestions should be addressed to:

Director
US Army Concepts Analysis Agency
ATTN: CSCA-MV
8120 Woodmont Avenue
Bethesda, MD 20814-2797

RESEARCH PAPER
CAA-RP-87-3

AN ALGORITHM FOR DETERMINING THE LOCATION OF POINTS IN COMPUTER SIMULATIONS

October 1987

DR. ROBERT L. HELMBOLD
OFFICE OF SPECIAL ASSISTANT
FOR MODEL VALIDATION

US Army Concepts Analysis Agency
8120 Woodmont Avenue
Bethesda, Maryland 20814-2797

DEPARTMENT OF THE ARMY
US ARMY CONCEPTS ANALYSIS AGENCY
8120 WOODMONT AVENUE
BETHESDA, MARYLAND 20814-2797

REPLY TO
ATTENTION OF

CSCA-MVM

19 FEB 1988

MEMORANDUM FOR: Deputy Under Secretary of the Army (Operations Research), Headquarters, Department of the Army, Washington, D.C. 20310

SUBJECT: Algorithm for Determining Locations in Computer Simulations

1. This paper describes a new and effective algorithm for determining in computer simulations the locations of points relative to prespecified boundary or phase lines. It was developed to satisfy a need that arises frequently in computer simulations of military operations where boundary or phase lines often designate regions that need to be treated differently for one reason or another. It's being provided to you in the expectation that you will find it useful in your work.

2. Questions or inquiries should be directed to our Office, Special Assistant for Model Validation, U.S. Army Concepts Analysis Agency (CSCA-MV), 8120 Woodmont Avenue, Bethesda, MD 20814-2797, (301) 295-1669.

E. B. VANDIVER III
Director

CAA	AN ALGORITHM FOR DETERMINING THE LOCATION OF POINTS IN COMPUTER SIMULATIONS	STUDY SUMMARY CAA-RP-87-3

THE REASON FOR PERFORMING THIS STUDY was to develop and document an improved algorithm for determining in computer simulations the locations of points relative to prespecified boundary or phase lines. The need for an improved algorithm was recognized when a previously-used method was discovered to lack sufficient generality.

THE PRINCIPAL FINDINGS are that a useful algorithm can be developed for determining in computer simulations the locations of points relative to prespecified boundary or phase lines. This algorithm applies to boundary and phase lines of very general shape and configuration, and so improves on those previously used in some simulations. It is based on the notion of winding number, which is used in the mathematical theory of complex variables.

THE MAIN ASSUMPTION is that the prespecified boundary or phase lines can adequately be approximated by a polygonal line, i.e., a finite number of connected straight line segments.

THE PRINCIPAL LIMITATIONS are that the boundary or phase lines need to be defined and entered into computer memory by manual methods--but this is true for other point-location algorithms, as well. Also, as in all point-location algorithms, points so close to a boundary line as to be affected by arithmetical roundoff errors may not be assigned to the correct zones.

THE SCOPE OF THE WORK is limited to finding an improved algorithm for determining the location of points relative to prespecified boundary or phase lines.

THE WORK WAS PERFORMED on the initiative of Dr. Robert L. Helmbold of the US Army Concepts Analysis Agency's Model Validation Office. It was reviewed by Dr. Daniel Willard of the Office, Deputy Under Secretary of the Army for Operations Research; Dr. Ralph Johnson, CSCA-FOT; COL Irving R. Schuetze, CSCA-RQN; and Mr. Dick Lester, CSCA-MV.

Tear-out copies of this synopsis are at back cover.

CONTENTS

FIGURES

TABLES

CHAPTER 1

EXECUTIVE SUMMARY

1-1. PROBLEM. Develop an algorithm which, given several prespecified boundary or phase lines in a plane and a point in the plane, will determine where the given point is located relative to the prespecified boundary or phase lines. For example, in Figure 1-1, the algorithm should determine whether the point is to the left or right of each boundary or phase line. (Note: that the boundary or phase lines are considered to be oriented as illustrated in Figure 1-1.) Such oriented boundary lines are necessary if "left" and "right" are to have meaningful definitions. For the case shown in Figure 1-1, point PAPA is to the left of boundary lines ALFA, BRAVO, and CHARLIE, and is to the right of boundary line DELTA.

1-2. BACKGROUND. The need for determining where a given point lies relative to a family of boundary lines arises frequently in computer simulations of military operations where boundary or phase lines often designate regions that need to be treated differently for one reason or another. For example, the boundary lines may mark the border between friendly and enemy territory, national boundaries, air defense identification zones, areas where special rules of engagement (such as free-fire or no-fire) apply, contaminated and uncontaminated regions, areas of differing mobility characteristics, unit boundaries, fire support coordination lines, and so forth. As Figure 1-1 illustrates, humans can easily make these determinations if provided with a graphical display. However, in military simulations, it often is the case that no graphical displays are conveniently available, or that all the operations are to take place within the computer simulation without human monitoring or intervention, or both. In such cases, it is necessary to provide the computer with an algorithm for determining where the given point is located relative to the boundary lines.

For instance, the Nuclear Fire Planning and Assessment Model (NUFAM) at the US Army Concepts Analysis Agency (CAA) employs a boundary line to mark the separation between friendly and enemy target elements. An inspection of the algorithm originally proposed for that purpose revealed that it was:

a. Narrowly applicable in the sense that it could give incorrect results unless the boundary curves were of a very simple shape and orientation.

b. Ad hoc in the sense of apparently not being based on any general principles.

c. User-hostile in the sense that its implementation involved a series of logical branches whose net result was not easily discernible. As a result, the algorithm appeared difficult to verify, debug, modify, or incorporate confidently into laser programs.

d. Expensive in the sense of requiring considerable computer time for determining the location of a given point relative to prespecified boundary or phase lines.

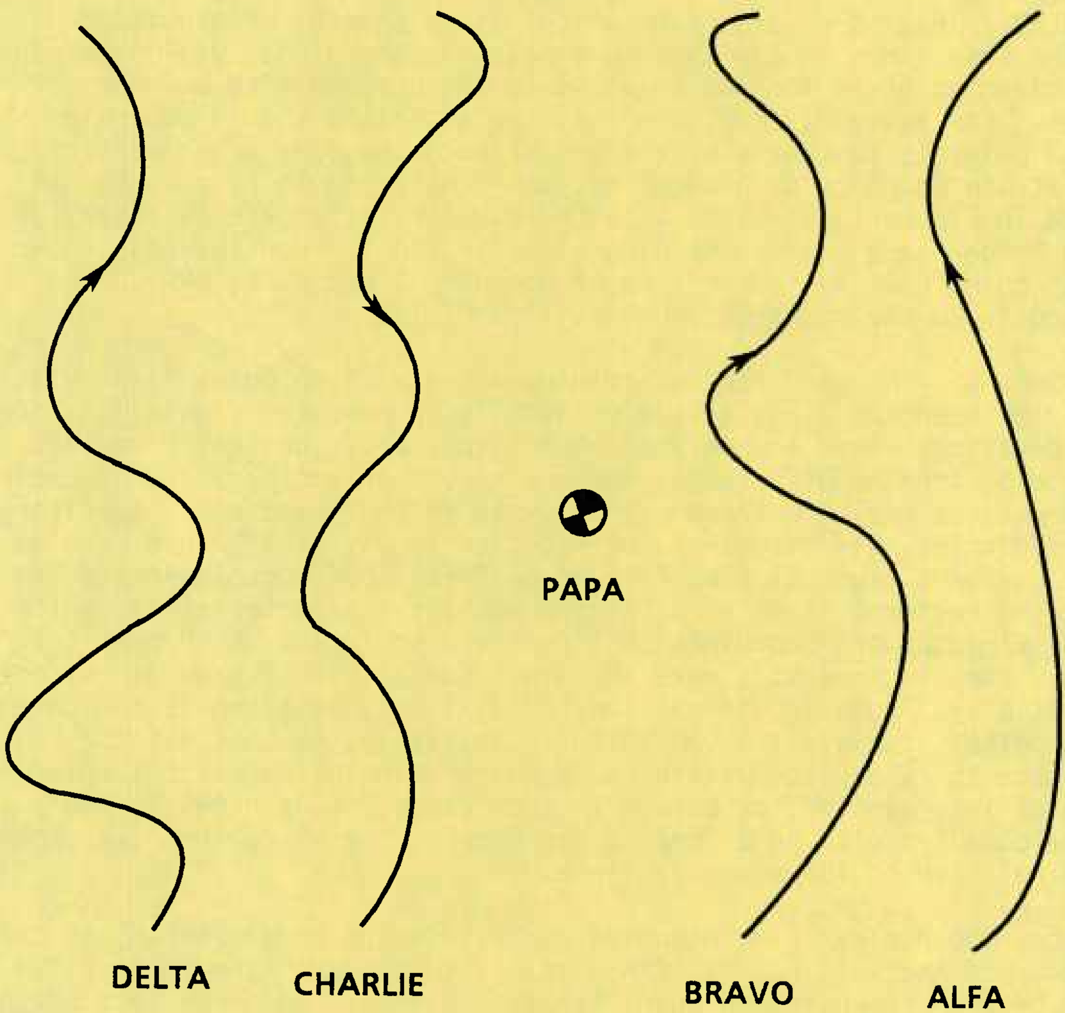

Figure 1-1. Point PAPA and Boundary Lines
ALFA, BRAVO, CHARLIE, and DELTA

1-3. SCOPE. This paper develops an algorithm for determining where a given point is located relative to prespecified boundary or phase lines. The algorithm presented here is:

a. Widely applicable in the sense that it gives correct answers even when the boundary curves are complex in shape and orientation.

b. Explicitly based on general principles.

c. User-friendly in the sense that its implementation involves a straightforward computation whose net result is easily discernible. As a result, the algorithm should be easy to verify, debug, modify, or incorporate confidently into larger programs.

It still requires a fair amount of computer time to determine the location of a point, but it is not expected to need more than the one originally proposed for use in NUFAM.

1-4. LIMITATIONS. The algorithm presented here is sometimes inconsistent in its assignment of points on (or quite close to) the boundary. That is, it may sometimes count these points to the left of the boundary when they actually are on or to the right of it, and vice versa. However, this phenomenon occurs only for points on or so close to the boundary that machine-specific limitations on arithmetic precision affect the results. Such limitations arise from inescapable restrictions on the number of significant figures used in the computations, and hence affect all such algorithms.

1-5. TIMEFRAME. Not applicable.

1-6. KEY ASSUMPTIONS. The key assumption is that the boundary curves can adequately be approximated by polygonal lines. The same assumption is used in NUFAM and in many other computer simulations. In this paper, we adopt the convention that all boundary or phase lines, all polygonal lines, and all closed curves (whether smooth or polygonal) are considered to be oriented. Henceforth, the fact that they are oriented will be mentioned only occasionally as a reminder or for emphasis. But the fact that they are oriented is always to be understood, even if it's not mentioned explicitly. In this paper, we adopt a further convention for boundary or phase lines separating friendly from enemy territory. This is, that friendly forces or territory always lies to the left of an oriented boundary or phase line, and enemy forces or territory to the right of it. Examples of how this is implemented are shown in Chapter 3.

1-7. APPROACH. The approach is based on the general theory of winding numbers developed for the mathematical theory of complex functions (see Ahlfors, pp 92-94, among others). The winding number of a prespecified (oriented) closed curve with respect to a given point is the number of times the prespecified closed curve winds (in a counterclockwise direction) about the given point. In Figure 1-2, the closed curve GOLF winds zero times about the point PAPA-0, once (counterclockwise) about the point PAPA-1, and twice (counterclockwise) about the point PAPA-2. To apply winding number theory to determining the location of points relative to an (oriented) boundary or phase line, we extend the boundary or phase line to form a closed (oriented) curve. The extension of a boundary or phase line to a closed curve can be done either before or after the boundary line is approximated by a polygonal line. Figure 1-3 shows the stages in going from an original boundary line, to an approximating polygonal one, to a closed polygonal curve. The approximating polygonal boundary line can be chosen in many different ways, and the extension to a closed polygonal curve can also be done in many different ways. Naturally, the analyst should choose a closed polygonal curve that facilitates subsequent analysis.

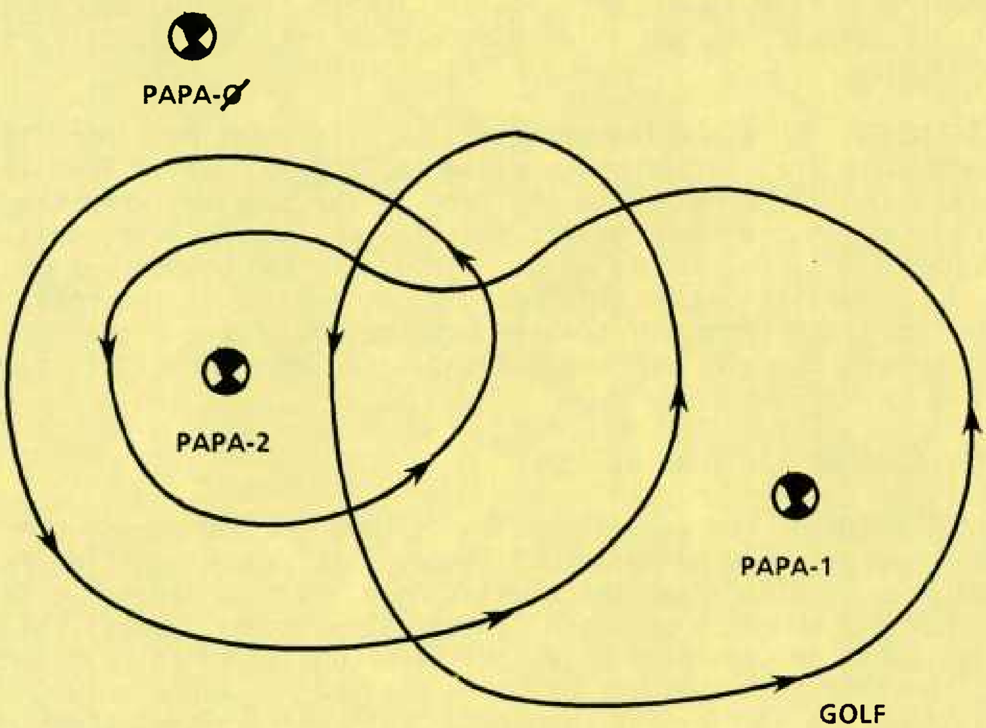

Figure 1-2. Closed Curve Golf Illustrating Winding Number With Respect to Points PAPA-0, PAPA-1, and PAPA-2

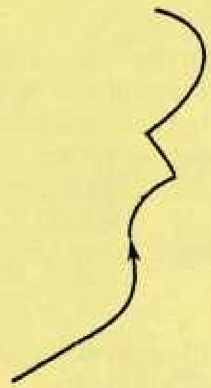

Figure 1-3a. Original Oriented Boundary Line

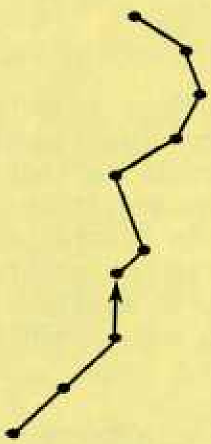

Figure 1-3b. An Oriented Polygonal Approximation To the Original Oriented Boundary Line

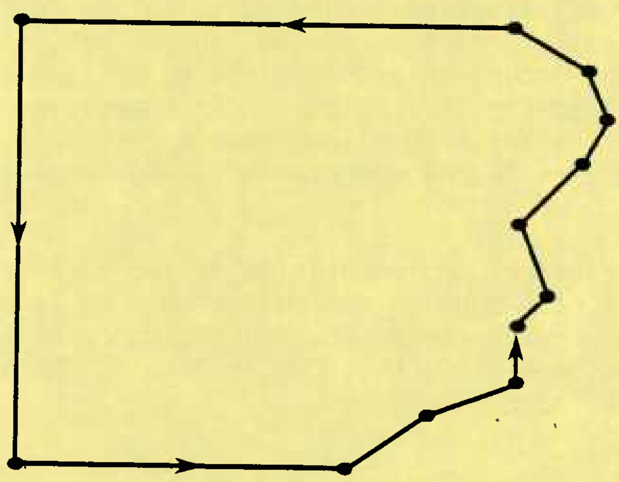

Figure 1-3c. An Oriented Closure Of An Approximating Polygonal Boundary Line

The benefits from replacing an original boundary line by a closed polygonal curve arise from the following facts:

a. Points to the left of the original boundary line are inside the closed polygonal curve, and hence have winding numbers of one with respect to it. Also, points to the right of the original boundary line are outside the closed polygonal curve, and hence have winding numbers of zero with respect to it. Hence, we have replaced the original problem of determining whether a given point is to the right or left of the boundary line with the problem of calculating the winding number of the closed polygonal curve with respect to the given point.

b. The winding number of a closed polygonal curve with respect to a point can be computed by a simple finite algorithm.

Hence, a simple finite algorithm suffices to determine whether a point is to the left or right of the original boundary line. The details of this development are provided in Chapter 2. Examples are presented in Chapter 3.

1-8. CONCLUSIONS

a. A simple finite algorithm suffices to determine the location of a point relative to (oriented) boundary or phase lines. The resultant algorithm is widely applicable in the sense that it gives correct answers even when the boundary curves are complex in shape and orientation. It has the merit of being based explicitly on general mathematical principles. It is user-friendly in the sense that its implementation involves a straightforward computation whose net result is easily discernible. As a result, the algoritihm should be easy to verify, debug, modify, or incorporate confidently into larger programs.

b. The algorithm still requires a fair amount of computer time to determine the location of a point, but it is not expected to need more than the one originally proposed for use in NUFAM. The algorithm also is sometimes inconsistent in its assignment of points on (or quite close to) the boundary. That is, it may sometimes count these points to the left of the boundary when they actually are on or to the right of it, and vice versa. However, this phenomenon occurs only for points on or so close to the boundary that machine-specific limitations on arithmetic precision affect the results. Such limitations arise from inescapable restrictions on the number of significant figures used in the computations, and hence affect all such algorithms.

1-9. OBSERVATIONS. Practical implementation of the algorithm would be aided by the development of a fast-running subroutine that could be incorporated into large simulations or wargames such as NUFAM, Concepts Evaluation Model (CEM), Force Evaluation Model (FORCEM), Combat Sample Generator (COSAGE), Vector in Commander (VIC) and others.

CHAPTER 2

APPROACH

2-1. APPROACH. Throughout the rest of this paper, it will be assumed that the original boundary or phase lines have been replaced by closed polygonal curves, as described in paragraph 1-7. As stated there, this replaces the original problem of finding whether a point is to the left or right of the original boundary line by one of finding the winding number of the closed polygonal curve with respect to the given point. This chapter outlines a procedure for obtaining the winding number of a closed polygonal curve with respect to a given point. A program to perform the necessary computations is provided in Appendix D. Examples of closed polygonal curves for use with this computer program are provided in Chapter 3.

2-2. NUMBER VERTICES. The first step is to number the vertices of the closed polygonal curve. Number them consecutively from one to N. Assign increasing numbers to vertices in order according to the orientation of the closed polygonal curve. In principle, the number "1" can be assigned arbitrarily to any vertex of the polygonal curve, but usually one choice is more convenient than the others. Since the choice does not matter to the theory, the analyst is free to choose the most convenient "number 1" vertex. Figure 2-1 shows how the 12 vertices of the closed polygonal curve from Figure 1-3 might be numbered.

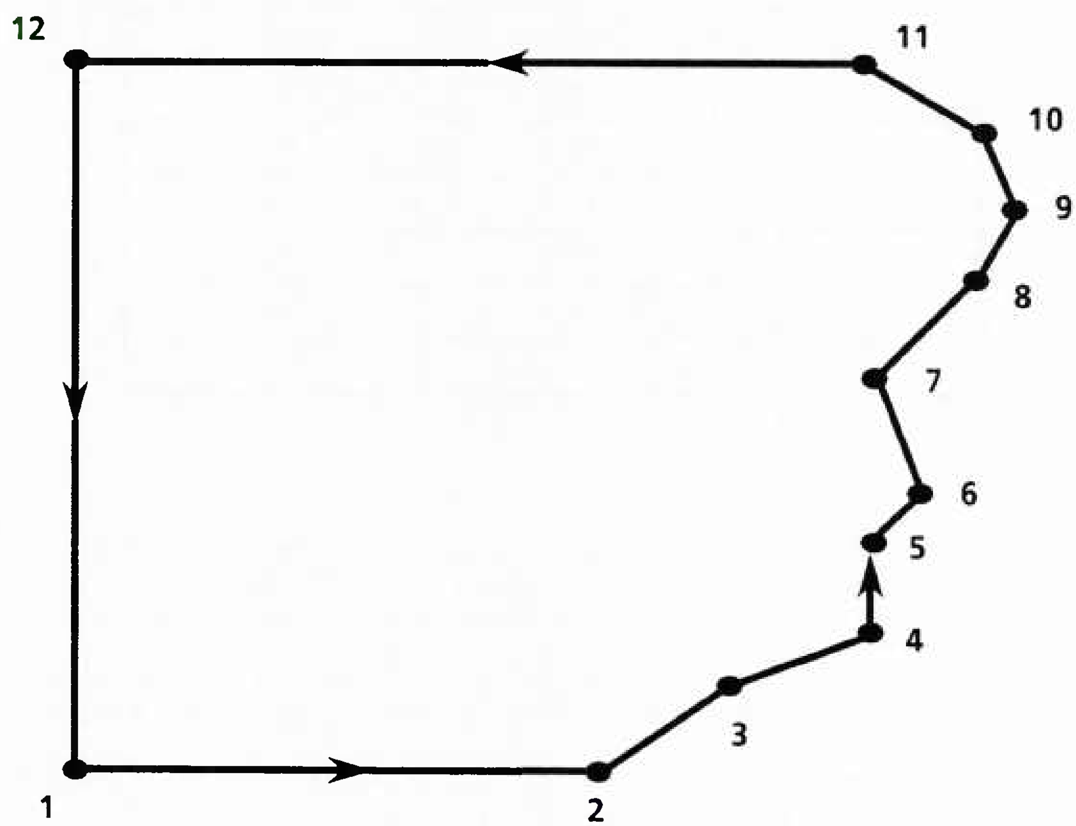

Figure 2-1. A Vertex Numbering for the Closed Polygonal Curve from Figure 1-3

2-3. TABULATE COORDINATES OF VERTICES. Make a table of the coordinates of the vertices of the closed polygonal curve. The rows of this table must correspond to the numbers assigned to the vertices. Table 2-1 shows how the table of coordinates might look for the closed polygonal curve shown in Figure 2-1.

Table 2-1. Coordinates of Vertices For The Closed Polygonal Curve in Figure 2-1

Vertex number	Coordinates	
	x	y
1	0	0
2	7.0	0
3	8.2	1.6
4	9.4	2.2
5	9.5	3.5
6	9.9	4.0
7	9.5	5.2
8	10.6	6.1
9	10.9	7.0
10	10.6	7.9
11	9.5	8.3
12	0	8.3

Observe that the notions of inside and outside are topological in nature. The use of this observation is that the coordinates of the vertices may be recorded in any convenient coordinate systems provided only that:

a. The coordinates of the given point are recorded in the same coordinate system as that used to record the vertices of the closed polygonal curve, and

b. There is a continuous (i.e., topological) mapping of the coordinates onto the region defined by the closed polygonal curve.

This freedom to choose any convenient coordinate system can be used by the analyst to choose one that makes his work easy. For example, the Universal Transverse Mercator (UTM) grid system could be used as the coordinate system.

Latitude and longitude could also be used, as could an arbitrary Cartesian coordinate system. Provided they are used consistently once chosen, results will be the same no matter which coordinate system is used. (Problems arise at discontinuities in the UTM grid, and at discontinuities in latitude-longitude at the poles and at the Prime Meridian. These difficulties arise because the mapping of the coordinates onto the region is not topological across UTM grid discontinuities or across latitude-longitude discontinuities. However, with the possible exception of very large expanses on the globe, the analyst will see how to choose a convenient coordinate system that is free of these discontinuities.)

2-4. OBTAIN COORDINATES OF GIVEN POINT. Obtain the coordinates of the given point relative to the same coordinate system used to record the locations of the vertices. For example, for the closed polygonal curve of Figure 2-1, the given point may be at $x = 5$ and $y = 5$. This point is obviously inside the closed polygonal curve, and hence is to the left of the boundary line.

2-5. FIND WINDING NUMBER. Use the algorithm of Appendix D to determine the winding number of the closed polygonal curve with respect to the given point. For the vertices shown in Table 2-1 and a given point at coordinate $x = 5$ and $y = 5$, the winding number will be equal to 1.

2-6. CONTINUATION. Repeat the steps described in paragraphs 2-3 and 2-4 until the winding numbers of all of the closed polygonal curves with respect to each given point have been determined.

2-7. CAUTIONARY NOTE. Avoid closed polygonal curves that do not separate the plane into well-defined "inside" and "outside" pieces. Figure 2-2 is an example of the type of closed polygonal curve to avoid. It is not clear whether the shaded region is intended to be considered "inside" or "outside" the closed curve, which therefore fails to separate the plane into clearly defined "inside" and "outside" components. In fact, the algorithm of Appendix D assigns a winding number of minus one to points in the heavily shaded region, which (given the conventions adopted in paragraph 1-6) would be interpreted as being in enemy territory. The algorithm assigns a winding number of plus one to points in the lightly shaded regions, and a winding number of zero to points in the unshaded region.

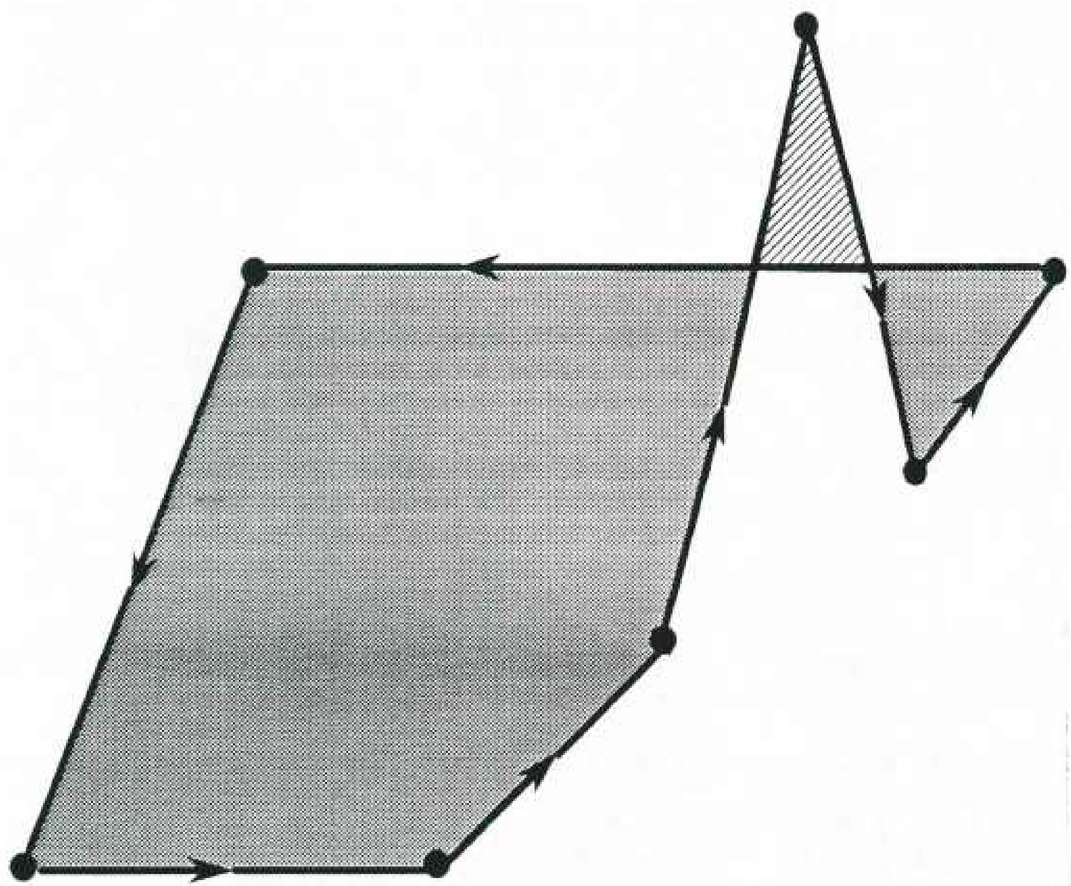

Figure 2-2. Example of a Closed Polygonal Curve to Avoid

2-8. MOVING BOUNDARY AND PHASE LINES. After units move, or boundary and phase lines shift, the location of the points representing unit position usually have to be recomputed. When movement is represented as a series of "snapshots" of an evolving situation, the same procedures as outlined above and illustrated in Chapter 3 can be applied at each "snapshot" to obtain the unit locations relative to pertinent boundary or phase lines.

CHAPTER 3

RESULTS

3-1. INTRODUCTION. This chapter presents some examples to illustrate the flexibility of the algorithm and the options it provides the analyst. The program in Appendix D correctly computes winding numbers for these and other cases that have been tried.

3-2. A SERPENTINE-SHAPED REGION. To test the algorithm, the serpentine-shaped region of Figure 3-1 was constructed. This shape has no obvious military tactical significance, but it is suitable for testing the operation of the algorithm. Note that Figure 3-1 shows the vertex numbers for only the first and last two or three vertices. This was done to avoid cluttering the figure with redundant symbols. The other vertices are, of course, numbered in order according to the orientation of the closed polygonal curve bounding the serpentine shaped region. The vertex coordinates are given in Table 3-1.

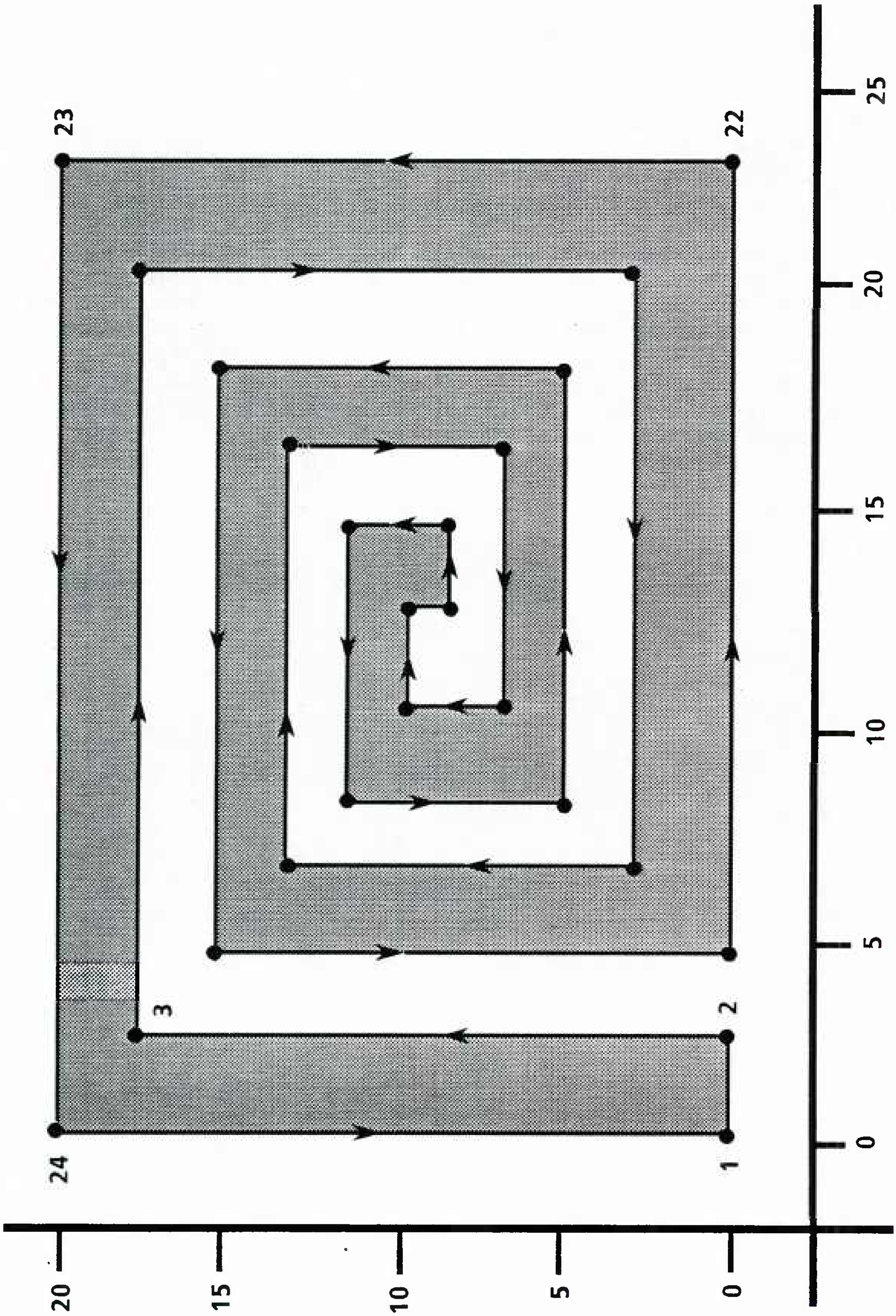

Figure 3-1. A Serpentine-Shaped Region

Table 3-1. Vertex Coordinates For Figure 3-1

Vertex number	Coordinates	
	x	y
1	0	0
2	2	0
3	2	18
4	20	18
5	20	2
6	6	2
7	6	14
8	16	14
9	16	6
10	10	6
11	10	10
12	12	10
13	12	8
14	14	8
15	14	12
16	8	12
17	8	4
18	18	4
19	18	16
20	4	16
21	4	0
22	22	0
23	22	20
24	0	20

When used with the algorithm of Appendix D, the points in the shaded region are assigned a winding number of plus one, as they should since they lie inside the closed polygonal curve. Points in the unshaded region are assigned a winding number of zero, as they should since they are outside the closed polygonal curve. Points on the closed polygonal curve may be assigned a winding number of zero or plus one, depending on the specific numerical rounding method employed by the particular computer machine used to perform the algorithm.

3-3. DISJOINT COMPONENTS. Figure 3-2 shows a shaded region consisting of two components bounded by a closed polygonal curve. Militarily the inner shaded area may represent a friendly force surrounded by enemy forces, who are themselves surrounded by a larger friendly force represented by the outer shaded area. Table 3-2 gives the vertex coordinates used in the algorithm of Appendix D. The winding number is plus one for points in either shaded region, and zero elsewhere.

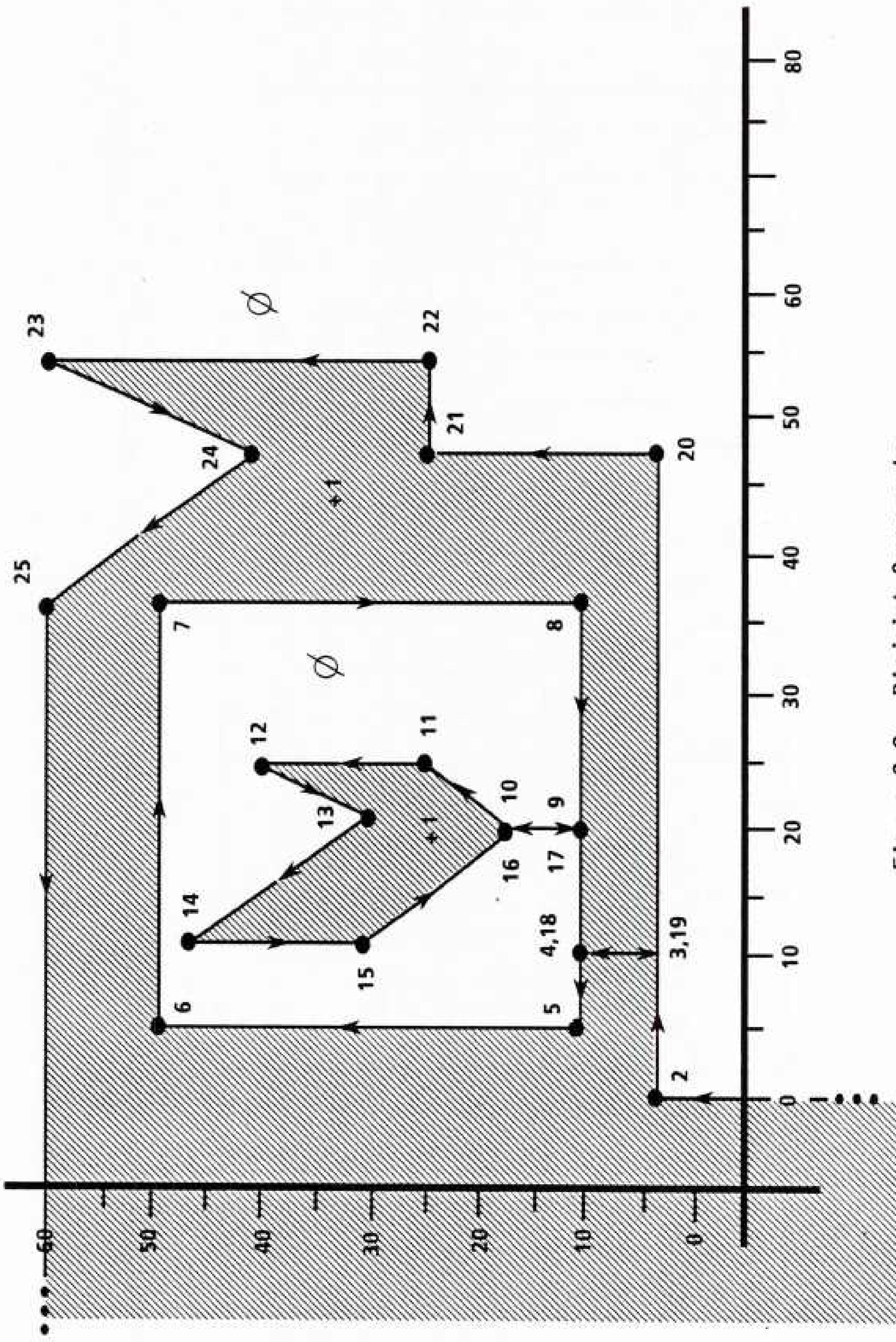

Figure 3-2. Disjoint Components

Table 3-2. Vertex Coordinates For Figure 3-2

Vertex number	Coordinates	
	x	y
1	0	-10^6
2	0	5
3	10	5
4	10	10
5	5	10
6	5	50
7	35	50
8	35	10
9	20	10
10	20	15
11	25	20
12	25	40
13	20	25
14	10	45
15	10	25
16	20	15
17	20	10
18	10	10
19	10	5
20	45	5
21	45	20
22	55	20
23	55	60
24	45	40
25	35	60
26	-10^6	60
27	-10^6	-10^6

a. One of the points of this example is that the algorithm will give the correct winding number for disjoint components, like those shown in Figure 3-2.

b. A second point is that this approach fails to distinguish points like the one at (100, 100) which are completely outside the shaded regions from points like the one at (30, 30) which are between the two shaded components. If that distinction is important, then separate closed polygonal curves can be used (e.g., one for the boundary of the inner shaded region, one for the inner and outer boundaries of the region between the two shaded components, and a third for the inner and outer boundaries of the outer shaded region).

c. A third point of this example is the closed polygonal curves may be strung together by links like those between vertices 9 and 10 that are traversed twice during the circuit around the polygonal curve. The fact that this link is traversed once from vertex 9 to vertex 10, and again in the reverse direction from vertex 16 to vertex 17, causes it to "cancel out" of the winding number computation. Points on this link are correctly assigned a winding number of zero. Similarly, points on the link from vertex 3 to vertex 4 are correctly assigned a winding number of plus one.

3-4. MULTIPLE ZONES. Figure 3-3 shows a closed polygonal curve that bounds multiple zones. This figure may correspond to a military situation where all territory to the left of boundary line A is held by friendly forces, territory between boundary lines A and AH is "no man's land," and territory to the right of boundary line AH is enemy-held territory. In addition, various zones in friendly territory are separated by phase lines B, C, and D. Similarly, zones in enemy-held territory are separated by phase line BEY.

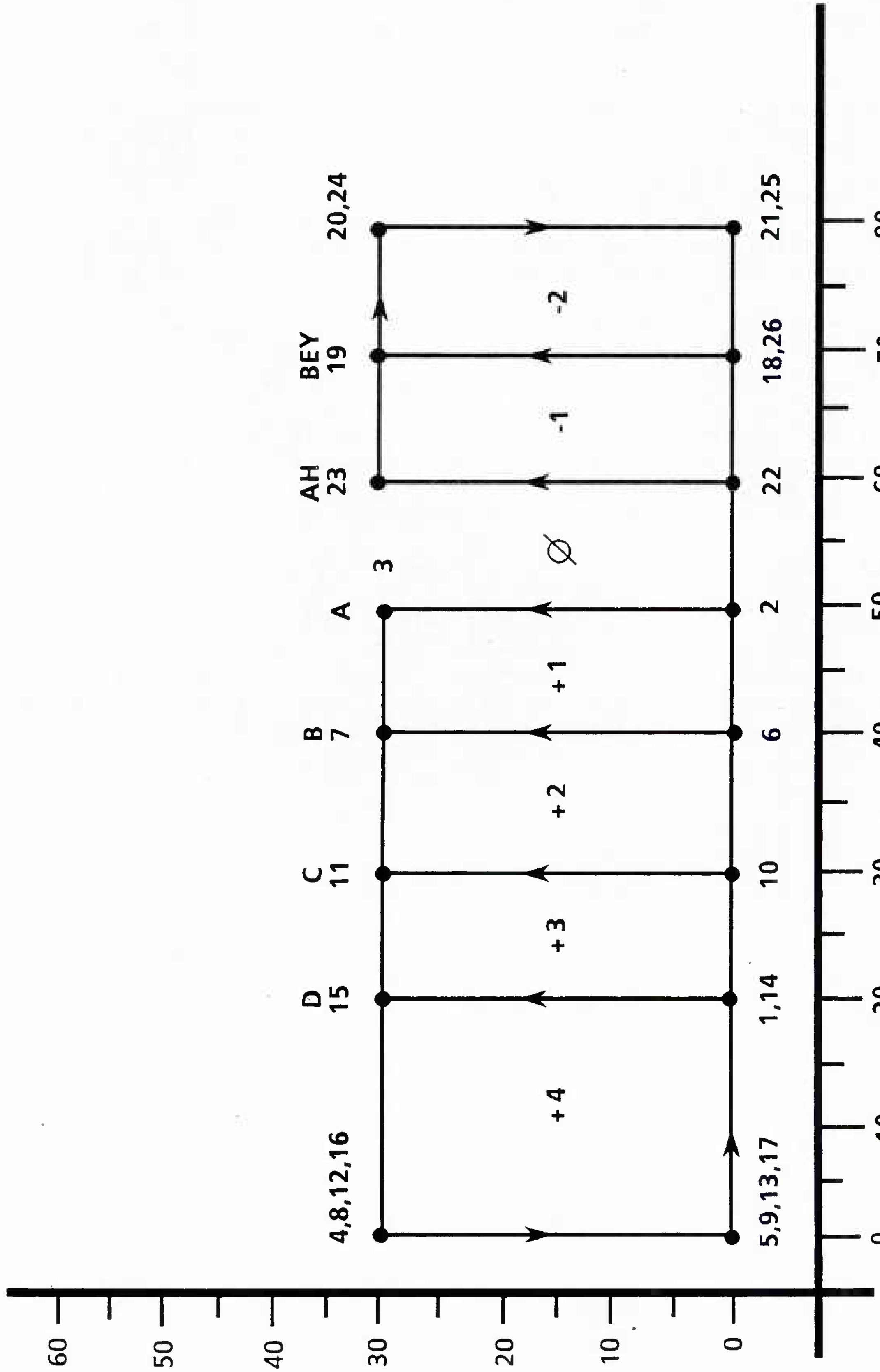

Figure 3-3. Multiple Zones

a. The vertex numbers in Figure 3-3 indicate the manner in which the curve is traversed. This manner of traversal is chosen in such a way that each zone has a different winding number. Negative winding numbers indicate that the curve winds about the point in a clockwise manner. Positive winding numbers indicate that the curve winds about the point in a counterclockwise manner. Note that the conventions adopted in paragraph 1-6 imply that positive winding numbers are used for zones in friendly territory, and negative winding numbers are used for zones in enemy-held territory. For example, points in the zone between phase lines A and B have a winding number of plus one, and points in the zone to the right of phase line BEY have a winding number of minus two.

b. Since the winding numbers uniquely identify the zones, to determine which zone a given point lies in it suffices to compute the winding number of the closed polygonal curve with respect to the given point.

c. Table 3-3 gives the coordinates of the vertices for Figure 3-3. As usual, some points on the boundary may be assigned to an unrelated zone, due to machine-specific details of roundoff operations. This is inherent in the lack of exact correspondence between theoretical mathematical operations and those actually carried out by computing machines, and hence unavoidable.

Table 3-3. Vertex Coordinates For Figure 3-3

Vertex number	Coordinates	
	x	y
1	20	0
2	50	0
3	50	30
4	0	30
5	0	0
6	40	0
7	40	30
8	0	30
9	0	0
10	30	0
11	30	30
12	0	30
13	0	0
14	20	0
15	20	30
16	0	30
17	0	0
18	70	0
19	70	30
20	80	30
21	80	0
22	60	0
23	60	30
24	80	30
25	80	0
26	70	0

3-5. RUNNING TIME CONSIDERATIONS. The program at Appendix B is written in APPLESOFT BASIC. Timing checks indicate that this program, when used on an APPLE II computer, takes approximately 0.01 seconds per side of the closed polygonal curve for each given point. Therefore, we can estimate that computing the winding numbers for each of 10,000 given points with respect to a closed polygonal curve having 100 sides would take 10,000 seconds, or 167 minutes on the APPLE II. If mainframe computers are 100 times faster, then they could do these computations in only about 1.7 minutes. The required time could be shortened if the winding number program were written to optimize its speed.

CHAPTER 4

CONCLUSIONS AND OBSERVATION

4-1. CONCLUSIONS

a. A simple finite algorithm suffices to determine the location of a point relative to (oriented) boundary or phase lines. The resultant algorithm is widely applicable in the sense that it gives correct answers even when the boundary curves are complex in shape and orientation. It has the merit of being based explicitly on general mathematical principles. It is user-friendly in the sense that its implementation involves a straightforward computation whose net result is easily discernable. As a result, the algorithm should be easy to verify, debug, modify, or incorporate confidently into larger programs.

b. The algorithm still requires a fair amount of computer time to determine the location of a point, but it is not expected to need more than the one originally proposed for use in NUFAM. The algorithm also is sometimes inconsistent in its assignment of points on (or quite close to) the boundary. That is, it may sometimes count these points to the left of the boundary when they actually are on or to the right of it, and vice versa. However, this phenomenon occurs only for points on or so close to the boundary that machine-specific limitations on arithmetic precision affect the results. Such limitations arise from inescapable restrictions on the number of significant figures used in the computations, and hence affect all such algorithms.

4-2. OBSERVATIONS. Practical implementation of the algorithm would be aided by the development of a fast-running subroutine that could be incorporated into large simulations or war games such as NUFAM, CEM, FORCEM, COSAGE, VIC and others.

APPENDIX A

REFERENCE

Ahlfors, Lars V., "Complex Analysis: An Introduction to the Theory of Analytic Functions of One Complex Variable," McGraw-Hill Book Company, Inc., New York, NY, 1953

APPENDIX B

COMPUTER PROGRAM

```
22000  REM ---FIND WINDING NUMBER
22005  A$ = "N"
22010  PI = 3.1415926536: TWOPI = 2 * PI
22020  HOME : PRINT "FIND WINDING NUMBER WRT COLS 1 AND 2" :  PRINT:  PRINT
22025  PRINT "NOTE: COLS 1 & 2 GIVE THE COORDS OF THE" : PRINT "VERTICES OF
         AN ORIENTED CLOSED POLYGON"
22027  PRINT
22030  IF A$ = "N" THEN INPUT "USE MATRIX NO. = "; M
22035  IF A$ <  > "N" THEN PRINT "USE MATRIX NO. = "; M
22040  PRINT : PRINT "ENTER COORDINATES OF THE BASE POINT:"
22050  INPUT "      X = "; X
22060  INPUT "      Y = "; Y
22110  AX = M(M,RR,1) - X
22120  AY = M(M,RR,2) - Y
22130  GOSUB 22900:  REM ---FIND ARG(AX,AY) = ANGLE OF VERTEX RR = VERTEX 0,
         AS VIEWED FROM THE BASE POINT
22150   A = 0:  REM ---INITIALIZE ACCUMULATOR TO ZERO
22160   FOR J = 1 TO RR:  REM ---PROCEED AROUND THE POLYGON IN ORDER OF ITS
         VERTICES
22165  XARG = ARG:  REM ---XARG = ARG OF THE PRECEDING VERTEX
22170  AX = M(M,J,1) - X
22180  AY = M(M,J,2) - Y
22185   GOSUB 22900:  REM ---FIND ANGLE OF VERTEX J AS VIEWED FROM THE
          BASE POINT
22190  DEXTA = ARG - XARG:  REM --- DEXTA = ANGLE SUBTENDED BY SIDE (J-1,J)
          AS VIEWED FROM THE BASE POINT
22200   A = A + DEXTA + TWOPI * ((DEXTA <  - PI) - (DEXTA> PI))
22210    NEXT
22220   OMEGA = A / TWOPI:  REM ---OMEGA = WINDING NUMBER
22230   OMEGA = INT (OMEGA + .1):  REM---ROUND TO NEAREST INTEGER
22270   PRINT  :  PRINT "WINDING NUMBER = "; OMEGA
22275  PRINT "WRT COLS 1 AND 2 OF MATRIX "; M
22277  PRINT  :  IF OMEGA = 0 THEN PRINT "POINT IS OUTSIDE"
22278   IF OMEGA>  = 1 THEN PRINT "POINT IS INSIDE"
22280  PRINT : PRINT "DO ANOTHER BASE POINT (Y/N) =  ";:  GET A$:  PRINT
22285  IF A$ <  > "N" THEN GOTO 22020
22299  RETURN
22900  REM ---FIND ARG(AX,AY)
22910  IF AX = 0 THEN ARG = SGN (AY) * PI / 2:  RETURN
22920  ARG = ATN (AY / AX) + PI * ((AX < 0) + 2 * (AX > 0) * (AY < 0)):
         RETURN
```

<u>PROGRAM NOTES</u>:

1. The above program is in APPLESOFT Basic.

2. ATN (AY/AX) = ARCTAN (AY/AX), and returns a value in radians between (- pi/2) and (+ pi/2).

3. Expressions like (AX < 0) are evaluated as one if the condition in parentheses is satisfied, and as zero otherwise.

APPENDIX C

DISTRIBUTION

Addressee	No of copies
Deputy Chief of Staff for Operations and Plans Headquarters, Department of the Army ATTN: DAMO-ZA Washington, DC 20310	1
Deputy Chief of Staff for Operations and Plans Headquarters, Department of the Army ATTN: DAMO-ZD Washington, DC 20310	1
Deputy Chief of Staff for Operations and Plans Headquarters, Department of the Army ATTN: DAMO-ZDF Washington, DC 20310	1
Deputy Chief of Staff for Logistics Headquarters, Department of the Army ATTN: DALO-ZA Washington, DC 20310	1
Deputy Chief of Staff for Logistics Headquarters, Department of the Army ATTN: DALO-PLF Washington, DC 20310	1
Commander US Army Logistics Center Fort Lee, VA 23801	1
Deputy Under Secretary of the Army (Operations Research) Washington, DC 20310	1

Addressee	No of copies
Assistant Secretary of the Army (Research, Development, and Acquisition) Washington, DC 20310	1
Director US Army TRADOC Analysis Center White Sands Missile Range, NM 88002	1
Director US Army Materiel Systems Analysis Activity ATTN: AMXSY-LM (Mr. Fox) Aberdeen Proving Ground, MD 21005-5071	1
Director US Army Ballistic Research Laboratories Building 305 Aberdeen Proving Ground, MD 21005	1
Commander Combined Arms Combat Development Activity Fort Leavenworth, KS 66027	1
Director Defense Nuclear Agency ATTN: LASS 6801 Telegraph Road Alexandria, VA 20305	1
Commander Army Research Institute 5001 Eisenhower Avenue Alexandria, VA 22333	1

Addressee	No of copies
Commander US Army Logistics Evaluation Agency New Cumberland Army Depot New Cumberland, PA 17070	1
Director Defense Logistics Studies Information Exchange US Army Logistics Management Center Fort Lee, VA 23801	1
Defense Technical Information Center ATTN: DTIC-DDA Cameron Station Alexandria, VA 22314-6145	2
US Army Nuclear and Chemical Agency 7500 Backlick Road, Bldg #2073 ATTN: Library Springfield, VA 22150	1
Commander Air Mobility Research and Development Laboratory Eustis Directorate Fort Eustis, VA 23604	1
Commander US Army Mobility Equipment Research and Development Command Fort Belvoir, VA 22060	1
Commander US Army Research, Development, and Acquisition Information Systems Agency Radford, VA 24141	1

Addressee	No of copies
The Pentagon Library (Army Studies Section) ATTN: ANRAL-RS The Pentagon Washington, DC 20310	1
Commander US Army Forces Command ATTN: AFOP-OM Fort McPherson, GA 30330	1
Director Program Analysis and Evaluation Office of the Secretary of Defense Room 2E330 The Pentagon Washington, DC 20310	1
Organization of the Joint Chiefs of Staff ATTN: J-8 Room 1D936, The Pentagon Washington, DC 20310	1
Joint Deployment Agency Deploying Systems Division MacDill Air Force Base, FL 33608	1
Commandant US Army War College Carlisle Barracks, PA 17013	1
Commandant US Army War College ATTN: Director, DMSPO Carlisle Barracks, PA 17013	1

Addressee	No of copies
Commandant US Army War College ATTN: Library Carlisle Barracks, PA 17013	1
Commandant US Army War College ATTN: CLW Carlisle Barracks, PA 17013	1
Center for Aerospace Doctrine, Research, and Education (AU) ATTN: CADRE/TA Maxwell Air Force Base, AL 36112-5532	1
Air University ATTN: AU CADRE/WGTA (Capt. Taylor) Maxwell Air Force Base, AL 36112-5000	1
Commandant US Air War College Maxwell Air Force Base, AL 36112-5532	1
Commandant US Navy War College Newport , RI 02840	1
President National Defense University ATTN: NDU-LD-CDC Washington, DC 20319-6000	1
Commandant Armed Forces Staff College Norfolk, VA 23511	1

Addressee	No of copies
Commandant US Army Command and General Staff College Fort Leavenworth, KS 66027	1
Commandant US Army Command and General Staff College ATTN: Department of Combat Development, Force Development Fort Leavenworth, KS 66027	1
Superintendent United States Military Academy ATTN: Mathematics Department West Point, NY 10996	1
Superintendent United States Military Academy ATTN: Engineering Department West Point, NY 10996	1
Superintendent Naval Postgraduate School Monterey, CA 93940	1
Naval Postgraduate School ATTN: Department of Operations Research Code 55PY (Professor Parry) Monterey, CA 93940	1
Commandant US Army Infantry School ATTN: ATSH-IVT Fort Benning, GA 31905	1

Addressee	No of copies
Commandant US Army Armor School ATTN: ATSB-CD Fort Knox, KY 40121-5215	1
Commandant US Army Field Artillery School Fort Sill, OK 73503	1
Commandant US Army Air Defense School Fort Bliss, TX 79916	1
Commandant US Army Aviation School Fort Rucker, AL 36360	1
Commandant US Army Engineer School Fort Belvoir, VA 22060	1
Commandant US Army Transportation School Fort Eustis, VA 23604	1
Commandant US Army Intelligence Center and School ATTN: ATSI-TD Fort Huachuca, AZ 85613	1
Commander in Chief United States Readiness Command ATTN: RCDA MacDill Air Force Base, FL 33608	1

Addressee	No of copies
Commander US Army Combat Developments Experimentation Command Fort Ord, CA 93941	1
CINCPAC Staff C3S ASII Box 32A Camp Smith, HI 96861	1
Commander US Army Western Command ATTN: APOP-SPM Fort Shafter, HI 96858-5100	1
Commander Eighth US Army APO San Francisco 96301	1
Commander US Army, Japan ATTN: AJCS APO San Francisco 96343	1
Deputy Chief of Staff for Intelligence Headquarters, Department of the Army Washington, DC 20310	1
Commander, USAITAC AIAIT-HI, Tech Info Bldg 203, STOP 314 Washington Navy Yard Washington, DC 20374	1

Addressee	No of copies
Commander/Director US Army Engineer Studies Center Casey Building, No. 2594 Fort Belvoir, VA 22060	1
Commander US Army Corps of Engineers 20 Massachusetts Avenue, NW Washington, DC 20314-1000	1
Commander US Army Missile Command Redstone Arsenal, AL 35898-5090	1
Commander US Army Ballistic Missile Defense Systems Command Huntsville, AL 35807	1
Commander in Chief US Army, Europe & Seventh Army ATTN: AEAGF APO New York 09403	1
Commander in Chief US Army, Europe & Seventh Army ATTN: AEAGX-OR (Dr. Leake) APO New York 09403	1
Commander US Army Training and Doctrine Command ATTN: ATCD-AU Fort Monroe, VA 23651	1

Addressee	No of copies
Commander US Army Training and Doctrine Command Fort Monroe, VA 23651	1
Commander US Army Materiel Command 5001 Eisenhower Avenue Alexandria, VA 22333	1
US Army CE Command Program Analysis and Evaluation Systems Analysis Division Fort Monmouth, NJ 07703	1
Air Force Center for Studies and Analyses AFCSA/SAMI Room 1D363, Pentagon Washington, DC 20330-5425	1
Headquarters, US Air Force Office of Worldwide Management of Studies & Analyses ATTN: AF/SAL The Pentagon Washington, DC 20330	1
Commander Tactical Air Command TAC Analysis Group Langley Air Force Base, VA 23665	1
Joint Studies Group (TAC) ATTN: OSS Langley Air Force Base, VA 23665	1

Addressee	No of copies
Commander USAF Systems Command Andrews Air Force Base Washington, DC 20334	1
Commandant Air Force Institute of Technology ATTN: AFIT-EN Wright-Patterson Air Force Base, OH 45433	1
Headquarters Rome Air Development Center (AFSC) ATTN: COAM (Mr. T. Humiston) Griffiss Air Force Base, NY 13441	1
Center for Wargaming Naval War College ATTN: Naval Warfare Gaming System Code 33 Newport, RI 02840	1
President Center for Naval Analyses 4401 Ford Avenue Post Office Box 16268 Alexandria, VA 22302-0268	1
Naval Research Laboratory ATTN: Code 5704 4555 Overlook Avenue Washington, DC 20375	1
Chief of Naval Operations ATTN: OP-96 Room 4A526, Pentagon Washington, DC 20350	1

Addressee	No of copies
Commander Naval Air Systems Command ATTN: Mission and Effectiveness Analysis Division (AIR-5264) Washington, DC 20361	1
Commander Naval Sea Systems Command Code 09B5 Washington, DC 20361	1
US Army Liaison Officer Naval Weapons Center China Lake, CA 93555	1
US LNO Supreme Allied Commander Atlantic Norfolk, VA 23511-5100	1
Marine Corps Operations Analysis Group Center for Naval Analyses 4401 Ford Avenue P. O. Box 11280 Alexandria, VA 22302-0268	1
Director USMC Development and Education Center Quantico, VA 22134	1
AFOTEC/OAN ATTN; Modelling Branch Kirtland Air Force Base, NM 87117	1
CINC, US REDCOM ATTN: RCSTA MacDill Air Force Base, FL 33608-6001	1

Addressee	No of copies
Internal Distribution:	
Unclassified Library	2
Dr. Robert L. Helmbold	16

GLOSSARY

ABBREVIATIONS, ACRONYMS, AND SHORT TERMS

CEM	Concepts Evaluation Model
COSAGE	Combat Sample Generator
FORCEM	Force Evaluation Model
NUFAM	Nuclear Fire Planning and Assessment Model
VIC	VECTOR in COMMANDER